CMPT 376W Journal Cyclomatic Complexity
By: Michael Chang   Date: February 28, 2021

Cyclomatic Complexity is a way for software engineer to calculate the complexity of their code base during code review. It is important for every software engineer to learn to use cyclomatic complexity because it allows them to find ways to reduce the complexity of the code. Also, it can help them to write cleaner code that is more maintainable, readable, and testable. First, it is important to understand the word cyclomatic. The prefix *cyclo-* came from the Latinized Greek word *kyklo* that means rotation or circle, while the suffix *-matic* came from the Greek word *matos* that means willing to performing something. Therefore, the word cyclomatic means willing to perform a rotation or a cycle. The term cyclomatic complexity is created in 1976 by a software engineer, Thomas J. McCabe, who found McCabe Software. McCabe idea is to use mathematical analysis to visualize the code base as a graph and use it to calculate the amount of complexity that is in the code. Since the developer's code can be expressed as a graph, they can use it to explain their code base's logic without writing any code in their documents. The complexity includes branching points, conditional statements (if, else, switch, and etc), and loops. Cyclomatic complexity can be calculating by connecting the ending point of the graph to the starting point and count the number of cycle formed from the graph.

Ex. Let's find the Cyclomatic Complexity of the function below

```
def sum(value1, value2):
        return value1 + value2
```

Draw the graph, it'll contain 3 nodes (starting point, return statement point, and ending point) connected by 3 edges. According to the graph, it is shaped into a ring, which formed a singe cycle. Therefore, the complexity of the function above is 1.
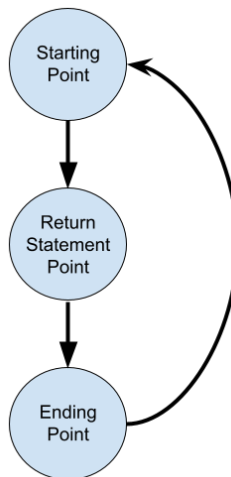
Cyclomatic complexity has a metric of scale, and the value will always be greater or equal to one. If the complexity is between 1 to 10, then it means that the code base is not complex. However, if the complexity is above 50, then it means that the code base is way too complex. This will cause the software application to be unstable.